Week 4 Lab - Sentiment Analysis II

Clarissa Boyajian

2022-05-03

```
## -- read in, clean, and wrangle data -- ##
files <- list.files(path = here("data/Week5"),
                    pattern = "pdf$", full.names = TRUE)

ej_reports <- lapply(files, pdf_text)

ej_pdf <- readtext(file = here("data/Week5", "*.pdf"),
                   docvarsfrom = "filenames",
                   docvarnames = c("type", "year"),
                   sep = "_")

# creating an initial corpus containing our data
epa_corp <- corpus(x = ej_pdf, text_field = "text" )

# context-specific stop words to stop word lexicon
more_stops <-c("2015","2016", "2017", "2018", "2019", "2020", "www.epa.gov", "https")
add_stops <- tibble(word = c(stop_words$word, more_stops))
stop_vec <- as_vector(add_stops)

# convert to tidy format and apply my stop words
raw_text <- tidy(epa_corp)

# Distribution of most frequent words across documents
raw_words <- raw_text %>%
  mutate(year = as.factor(year)) %>%
  unnest_tokens(word, text) %>%
  anti_join(add_stops, by = 'word') %>%
  count(year, word, sort = TRUE)

# number of total words by document
total_words <- raw_words %>%
  group_by(year) %>%
  summarize(total = sum(n))

report_words <- left_join(raw_words, total_words)

paragraph_tokens <- unnest_tokens(raw_text,
                            output = paragraphs, input = text,
                            token = "paragraphs")

paragraph_tokens <- paragraph_tokens %>%
 mutate(par_id = 1:n())
```

```
paragraph_words <- unnest_tokens(paragraph_tokens,
                                 output = word, input = paragraphs,
                                 token = "words")
```

Question 1

What are the most frequent trigrams in the dataset? How does this compare to the most frequent bigrams? Which n-gram seems more informative here, and why?

```
# clean tokens
tokens <- tokens(epa_corp, remove_punct = TRUE) %>%
  tokens_select(min_nchar = 3) %>%
  tokens_tolower() %>%
  tokens_remove(pattern = (stop_vec))
doc_freq_matrix <- dfm(tokens)
```

```
# bigrams
tokens_2 <- tokens_ngrams(tokens, n = 2)
doc_freq_matrix_2 <- dfm(tokens_2) %>%
  dfm_remove(pattern = c(stop_vec))

freq_words2 <- textstat_frequency(doc_freq_matrix_2, n = 20)
freq_words2$token <- rep("bigram", 20)

freq_words2
```

```
##                     feature frequency rank docfreq group  token
## 1     environmental_justice       556    1       6   all bigram
## 2      technical_assistance       139    2       6   all bigram
## 3            drinking_water       133    3       6   all bigram
## 4             public_health       123    4       6   all bigram
## 5           progress_report       108    5       6   all bigram
## 6               air_quality        73    6       6   all bigram
## 7             water_systems        66    7       6   all bigram
## 8    vulnerable_communities        65    8       6   all bigram
## 9                epa_region        62    9       5   all bigram
## 10     environmental_public        57   10       6   all bigram
## 11         federal_agencies        56   11       6   all bigram
## 12   national_environmental        51   12       6   all bigram
## 13           justice_fy2017        51   12       1   all bigram
## 14          fy2017_progress        51   12       1   all bigram
## 15          superfund_sites        48   15       4   all bigram
## 16       indigenous_peoples        46   16       6   all bigram
## 17             civil_rights        46   16       5   all bigram
## 18        local_governments        45   18       6   all bigram
## 19             urban_waters        44   19       6   all bigram
## 20 overburdened_communities        43   20       6   all bigram
```

```
# trigrams
tokens_3 <- tokens_ngrams(tokens, n = 3)
doc_freq_matrix_3 <- dfm(tokens_3) %>%
  dfm_remove(pattern = c(stop_vec))

freq_words_3 <- textstat_frequency(doc_freq_matrix_3, n = 20)
freq_words_3$token <- rep("trigram", 20)

freq_words_3
```

```
##                              feature frequency rank docfreq group   token
## 1            justice_fy2017_progress        51    1       1   all trigram
## 2             fy2017_progress_report        51    1       1   all trigram
## 3        environmental_public_health        50    3       6   all trigram
## 4       environmental_justice_fy2017        50    3       1   all trigram
## 5     national_environmental_justice        37    5       6   all trigram
## 6       office_environmental_justice        32    6       6   all trigram
## 7        epa’s_environmental_justice        32    6       6   all trigram
## 8     environmental_justice_progress        30    8       4   all trigram
## 9            justice_progress_report        30    8       4   all trigram
## 10    environmental_justice_concerns        30    8       5   all trigram
## 11            drinking_water_systems        29   11       5   all trigram
## 12      annual_environmental_justice        27   12       5   all trigram
## 13    environmental_justice_advisory        27   12       6   all trigram
## 14       fiscal_annual_environmental        25   14       3   all trigram
## 15          justice_advisory_council        24   15       6   all trigram
## 16      environmental_justice_grants        22   16       5   all trigram
## 17  technical_assistance_communities        20   17       6   all trigram
## 18 communities_environmental_justice        20   17       5   all trigram
## 19               safe_drinking_water        19   19       5   all trigram
## 20     technical_assistance_services        19   19       5   all trigram
```

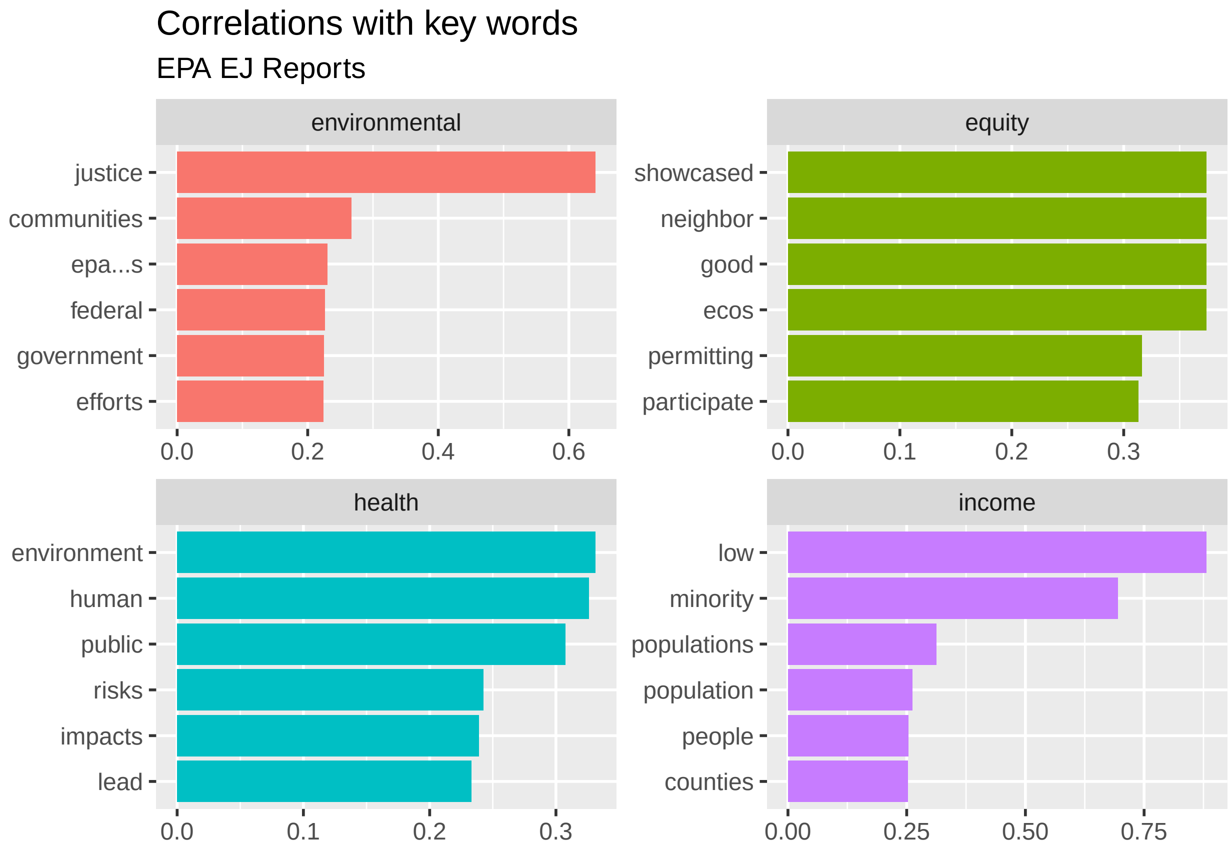
Answer: The trigrams appear to be less informative than the bigrams. Many of the trigrams include repetitive information with 6 of the top 10 including the phrase “environmental justice” with another, less inpactful word. Whereas the bigrams seem to include more individual topics, such as “public health”, “air quality”, and “vulnerable communities”.

Question 2

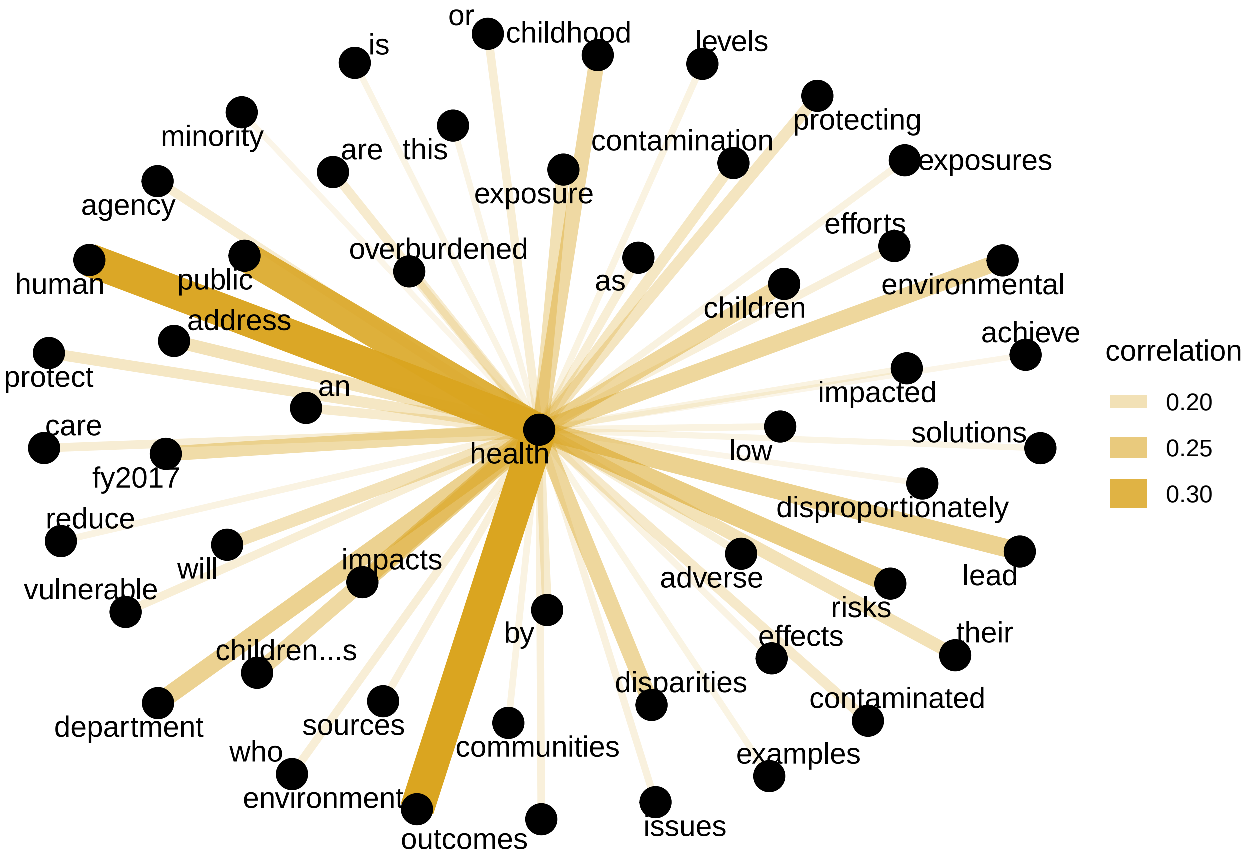
Choose a new focal term to replace “justice” and recreate the correlation table and network (see corr_paragraphs and corr_network chunks). Explore some of the plotting parameters in the cor_network chunk to see if you can improve the clarity or amount of information your plot conveys. Make sure to use a different color for the ties!

```
# word correlations
word_cors <- paragraph_words %>%
  add_count(par_id) %>%
  filter(n >= 50) %>%
  select(-n) %>%
  pairwise_cor(word, par_id, sort = TRUE)
```

```
# words correlated with "environmental", "health", "equity", and "income"
corr_table_data <- word_cors %>%
  filter(item1 %in% c("environmental", "health", "equity", "income")) %>%
  group_by(item1) %>%
  top_n(6) %>%
  ungroup() %>%
  mutate(item1 = as.factor(item1),
         name = reorder_within(item2, correlation, item1))

ggplot(data = corr_table_data,
       aes(y = name, x = correlation, fill = item1)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~item1, ncol = 2, scales = "free")+
  scale_y_reordered() +
  labs(y = NULL,
       x = NULL,
       title = "Correlations with key words",
       subtitle = "EPA EJ Reports")
```



```r
health_cors <- word_cors %>%
  filter(item1 == "health") %>%
  mutate(n = 1:n()) # add column that goes 1 to max rows
                    # (added to column that is ordered highest to lowest for correlation)

health_cors  %>%
  filter(n <= 50) %>% # get top 50 correlated words
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation), edge_colour = "goldenrod") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  theme_void()
```

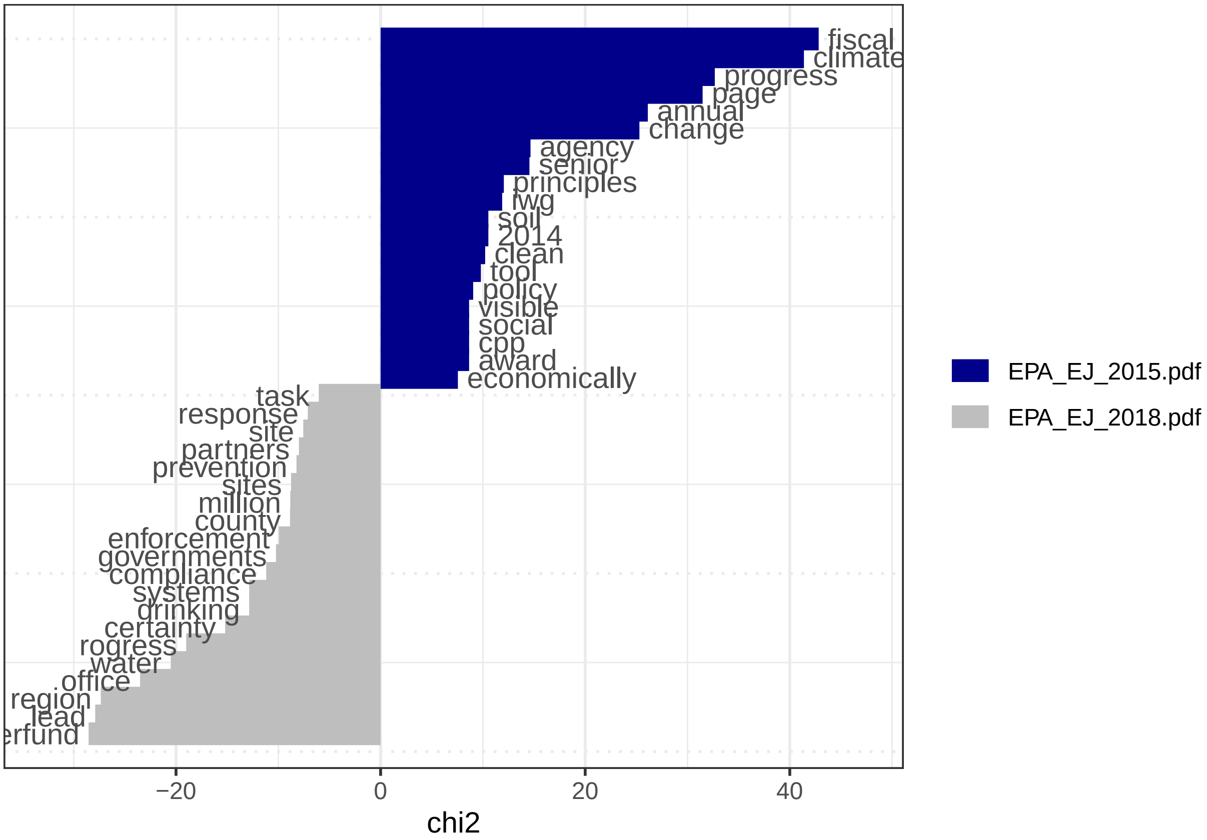


Question 3

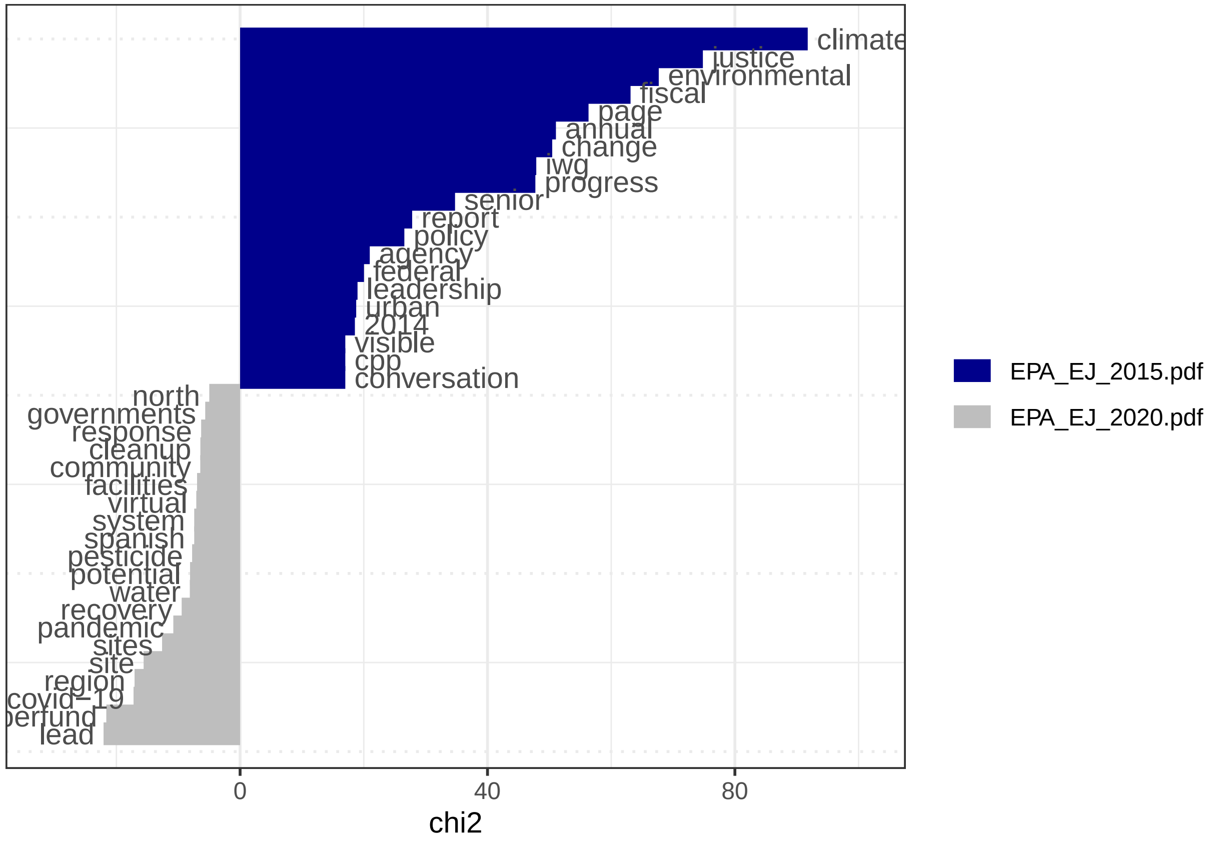
Write a function that allows you to conduct a keyness analysis to compare two individual EPA reports (hint: that means target and reference need to both be individual reports). Run the function on 3 pairs of reports, generating 3 keyness plots.

```r
keyness_plots <- function(years, target = 1){
  # create corpus based on input files
  files <- list.files(path = here("data/Week5"),
                      pattern = "pdf$", full.names = TRUE)

  ej_reports <- lapply(files, pdf_text)

  ej_pdf <- readtext(file = here("data/Week5", "*.pdf"),
                     docvarsfrom = "filenames",
                     docvarnames = c("type", "year"),
                     sep = "_") %>%
    filter(docvar3 %in% years)

  # creating an initial corpus containing our data
  epa_corp <- corpus(x = ej_pdf, text_field = "text" )

  tokens <- tokens(epa_corp, remove_punct = TRUE) %>%
    tokens_select(min_nchar = 3) %>%
    tokens_tolower() %>%
    tokens_remove(pattern = (stop_vec))

  doc_freq_matrix <- dfm(tokens)

  keyness <- textstat_keyness(doc_freq_matrix,
                              target = target) # target = 1 (refers to first document)
  textplot_keyness(keyness)
}
```

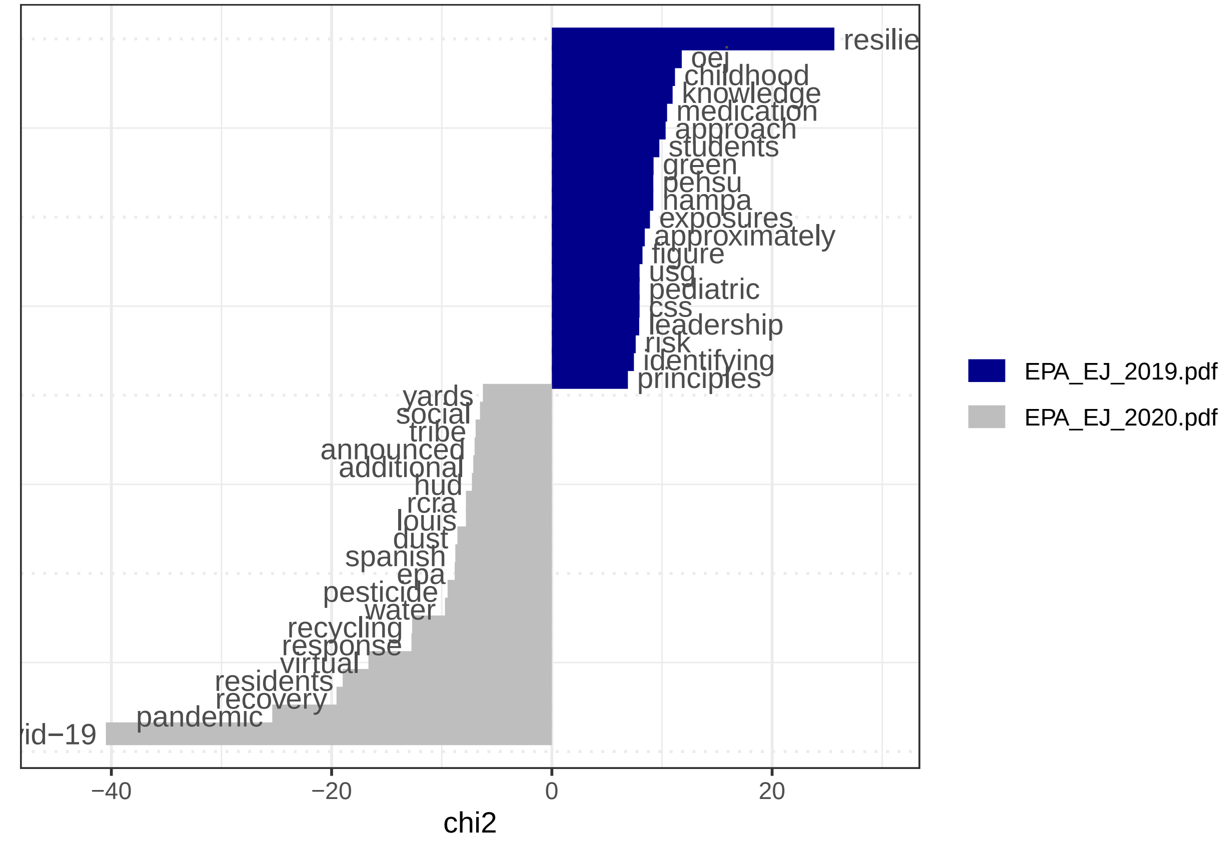
```r
# keyness plot comparing 2 years
keyness_plots(years = c(2015, 2018), target = 1)
```



```r
# keyness plot comparing 2 years
keyness_plots(years = c(2015, 2020), target = 1)
```



```
# keyness plot comparing 2 years
keyness_plots(years = c(2019, 2020), target = 1)
```



Question 4

Select a word or multi-word term of interest and identify words related to it using windowing and keyness comparison. To do this you will create two objects: one containing all words occurring within a 10-word window of your term of interest, and the second object containing all other words. Then run a keyness comparison on these objects. Which one is the target, and which the reference? Hint

```
term <- c("public", "health", "public health")

tokens_inside <- tokens_keep(tokens, pattern = term, window = 10) %>%
  tokens_remove(pattern = term) # remove the keywords

tokens_outside <- tokens_remove(tokens, pattern = term, window = 10)
```

```
doc_freq_matrix_inside <- dfm(tokens_inside)
doc_freq_matrix_outside <- dfm(tokens_outside)

tstat_key_inside <- textstat_keyness(rbind(doc_freq_matrix_inside, doc_freq_matrix_outside),
                                     target = seq_len(ndoc(doc_freq_matrix_inside)))
head(tstat_key_inside, 20)
```

```
##         feature      chi2            p n_target n_reference
## 1  environment 127.25447 0.000000e+00       72          36
## 2        human  99.34359 0.000000e+00       45          15
## 3      impacts  61.96533 3.441691e-15       57          50
## 4     meetings  51.48814 7.203127e-13       34          21
## 5  disparities  43.80985 3.618783e-11       22           9
## 6   children’s  42.86770 5.856959e-11       18           5
## 7         care  35.98937 1.983967e-09       24          15
## 8        risks  34.92443 3.427538e-09       25          17
## 9      comment  34.89753 3.475206e-09       11           0
## 10   exposures  34.47767 4.311690e-09       22          13
## 11  protecting  34.18740 5.005192e-09       18           8
## 12   childhood  28.17249 1.109708e-07       23          18
## 13     effects  26.93788 2.101010e-07       14           5
## 14    comments  26.58914 2.516547e-07       13           4
## 15     adverse  26.39516 2.782363e-07       12           3
## 16     improve  24.57389 7.151431e-07       42          57
## 17  department  23.80923 1.063711e-06       42          58
## 18      county  22.98673 1.631237e-06       26          27
## 19  experience  22.13078 2.546935e-06       16          11
## 20    distress  20.66615 5.467417e-06        7           0
```

Answer: The target is the list of all words within the 10 word window based on the key terms of “public health”. And the reference is the list of all other words in the EPA reports.